

## AMENDMENTS TO SPECIFICATION

Page 1, line 10 to Page 2, line 3:

A conventional method of testing server comprises connecting a great number of clients (i.e., computers) (e.g., 100 clients) to a server to be tested, and running a test program common to all clients for testing the server. Unfortunately, there are a plurality of test programs installed in each client. Thus, a test person has to watch carefully each test program selected sequentially by all clients in order to ensure that a test program common to all clients is selected by all clients in testing the server. However, human error is not avoidable since the selection is done manually. A correct test result is impossible ~~of obtaining~~ to obtain if a test program inputted by one client (or test programs inputted by two clients) is (or are) erroneous. Moreover, an immediate next test program cannot be displayed on the client after finishing the previous one. In this regard, the test person must pay a great attention to the process of each test program in each client. Unfortunately, a number of unexpected conditions may occur. For example, a test person forgets the finished percentage of a test program after leaving temporarily (e.g., going to bathroom). Alternatively, a test person is not aware of the finished test program. As such, a timing selection of an immediate next test program is not performed. As an end, one or more important test programs are not tested due to carelessness. Unfortunately, the test person has to perform the whole testing procedure again if such carelessness does occur. This can bring a great inconvenience to the test person as well as consume precious time and labor. Hence, a need for improvement exists.

Page 4, lines 3-13:

Referring to FIGS. 2, 3, and 4, there is shown a method in accordance with a preferred embodiment of the invention. The method first writes a plurality of basic databases and a window 20 into a memory of each of a plurality of clients in which each database comprises comparative data between client names and client IP, comparative data between test program names and test programs, and window picture and color of each test program. The window 20 comprises a client name field 21 for permitting a test person to input a client name, a client IP field 23, a test

program dialog box (e.g., Select NOS) ~~25~~ containing a plurality of (three) test program option buttons ~~of test program 25~~, a plurality of fields 26 of test ~~item~~ items and techniques, a test launch button 27, a set default gateway button 28, and a ping server button 29 for detecting a remote computer.

Page 4, line 14 to Page 5, line 11:

When the test person couples the clients to the server for testing the server, each client will read a window 20 from its memory prior to showing the same on the display. The test person then inputs client names on the client name field 21 of the window 20 by means of an input device. The client will search a matched IP from comparative data between the client names in memory and the client IP based on input client name. The found IP will be written into a register prior to showing on the client IP field 23 of the window 20. Next, the test person use an input device to click one of three option buttons 25 (e.g., Novell Netware, Windows NT, and SCO UnixWare) regarding a test program item to be tested in the dialog box ~~25~~ of the window 20. Next, the client searches a matched test program from comparative data between the test program names in memory and the test programs based on the selected test program item. The found test program will be written into a register. Next, read a corresponding color (e.g., red) 211 and a window picture 210 from the memory based on the selected test program item. Next, show the color 211 and the window picture 210 on the test program dialog box including test program option buttons 25 and one side of the test program dialog box ~~25~~ respectively. As such, it is possible ~~of giving to give~~ prominence to the selected test program item by means of the color 211 and the window picture 210. Hence, it will be easy for the test person to click a test item to be tested, a server name, a user ID, a client number, and a controller (e.g., automatic or manual) from the fields 26 of test ~~item~~ items and techniques. Each client will receive the test finish signal sent from the server after the server has finished the running of the test program. Next, show a window picture 210 and a color (e.g., blue) 211 of a next test program on a position on the window 20 corresponding to the next test program in the test program dialog box 25 (see FIG. 3). As such, the test person may know the finished percentage of a test program.

Serial Number 10/648,334

Page 5, lines 18-22:

Step 102: Click one of three test program option buttons (e.g., Novell Netware, Windows NT, and SCO UnixWare) regarding a test program item to be tested in the test program dialog box containing test program option buttons 25. Next, search a matched test program from comparative data between the test program names in memory and the test programs based on the selected test program item.

Page 5, lines 25-27:

Step 104: Write the found test program into a register. Next, show the color 211 and the window picture 210 on the test program dialog box containing test program option buttons 25 and one side of the test program dialog box 25-respectively for ease of being visually identified.

Page 6, lines 14-20:

Referring to FIG. 5, there is shown another preferred embodiment of the invention. The window 20 of each client further comprises a remote run dialog box 51. As such, the test person may select one of a plurality of ~~client numbers~~ remote run option buttons 52 and one of a plurality of commands 53 contained in the remote run dialog box 51 by clicking by operating any client. As a result, the test person can operate any of other remote clients for causing the remote client to run the selected command 53 based on IP.